I Claim:

1. A method for creating a GUI object for a hypertext application at a station, which comprises, the steps of:

reading a control element from a hypertext file;

parsing the control element into a set of attributes;

passing the parsed set of attributes to an object module; and,

receiving from the object module a GUI object of the control element.

2. A method as in claim 1 wherein the GUI object received contains the set of attributes.

3. A method as in claim 1 wherein the parsing step, further comprises, the following steps:

receiving a coordinate of the control element from an actuating device; and

determining a character offset of the coordinate of the control element.

4. A method as in claim 3 wherein the parsing step, further comprises, the steps of:

receiving a text font characteristic of the control element; and

organizing a leaf node for the text font characteristic of the control element.

5. A method as in claim 4 wherein the parsing step, further comprises, the step of indexing a pointer for the coordinate and the leaf node.

6. A method as in claim 5 wherein the parsing step, further comprises, the step of creating a leaf node for the parsed control element.

7. A method as in claim 1 which further comprises the steps of:

receiving from a text parser the parsed set of attributes corresponding to the control element; and

creating a GUI object of the control element using the parsed set of attributes.

8. A method as in claim 1, wherein the control element comprises a prompt box control.

9. A method as in claim 1, wherein the control element comprises a pop-up menu control.

10. A method as in claim 1, wherein the control element comprises a check-box control.

11. A method as in claim 1, wherein the control element comprises a calendar selector control.

12. A method as in claim 1, wherein the control element comprises a number selector control.

13. A method for displaying within a browser container a GUI object for a user, comprising, the steps of:

receiving a position location from an actuating device implemented by the user at a station;

mapping the position location to a corresponding leaf node having a GUI object associated with the leaf node; and,

displaying the GUI object to the user at the position location.

14. A method as in claim 13, wherein the station is connected across a distributed computer network.

15. A browser template for hypertext applications represented as data to a user located at a station, comprising:

a template defining an arrangement of control elements to be included in the browser template;

an object module containing information relating to a set of GUI objects of the control elements; and

a software engine which responds to text submitted by the user at the station by accessing a database and populating the editing template with a GUI object of the control element.

16. A method for activating an object module in an application operating on a hypertext document containing a control element, the control element having attributes, at a station which comprises the steps of:

parsing the attributes of the control element;

sending the attributes to an object module associated with the control element;

invoking the object module to make a GUI object containing the attributes;

activating the object module associated with the control element; and

displaying the GUI object.

17. The method as in claim 16, the sending step, which further comprises, the step of:

selecting a type of object module using a type of the control element.

18. The method as in claim 17, wherein a user activating an actuating device at a screen position corresponding to the control element incites the invoking, activating and displaying steps.

19. The method as in claim 17, wherein the control element has a corresponding displayed text label and an actuating device initiates the invoking, activating and displaying steps by a user activating the corresponding displayed text label.

20. The method as in claim 16 further comprising the step of activating a function corresponding to a type of object module invoked.

21. The method as in claim 20 wherein the function provides a user a selection of alternative inputs to the application.

22. The method as in claim 16, 17 or 18 wherein a mapping indexes a screen location of the parsed control element with a pointer to a corresponding object module.

23. A method as claimed in claim 16, 17 or 18, wherein each control element has a corresponding text label, which further comprises, the step of permitting a user to replace the corresponding control text element label with text typed in a GUI window.

24. A method as claimed in claim 16, 17 or 18, wherein each control element has a corresponding text label, which further comprises, the step of permitting a user to replace the corresponding control element label with text items selected from a set of displayed text items within a GUI window.

25. The method as claimed in claim 24, wherein the displayed text items are attributes of the control element.

26. A method as claimed in claims 16, 17 or 18, wherein each control element has a corresponding text label which further comprises, the step of permitting a user

to replace the corresponding control element label with items selected within a list of items using check boxes.

27. A method as claimed in claims 16, 17 or 18, wherein each control element has a corresponding text label, which further comprises, the step of permitting a user to replace the corresponding control element label with the item selected from a pop-up menu.

28. A method for editing text contained within an object module that is dynamically created in response to a control element in a hypertext document at a station, which comprises the steps of:

parsing the control element into a set of attributes;

displaying in an edit box an attribute, the attribute being editable, from the parsing step;

locating an object module corresponding to the edited attribute using a mapping from a location of the control element to a pointer to the object module corresponding to the edited attribute; and,

sending the edited attribute to the corresponding object module.

29. The method as in claim 16, wherein a mapping indexes a screen location of the parsed control element with a pointer to the corresponding object module.

30. A method as in claim 16, 17 or 18, wherein an attribute of a control element is a control element.

31. A method as in claim 30, which further comprises the steps of:

pushing the object module onto a stack; and

parsing the control element, the control element being the attribute of the control element corresponding to the pushed object module.

32. The method as in claim 30, wherein a mapping indexes a screen location of the parsed control element with a pointer to its corresponding object module.

33. A method as in claim 16 wherein the parsing step, further comprises, the following steps:

receiving a coordinate of the control element from an actuating device; and,

determining a character offset of the coordinate of the control element.

34. A method as in claim 17, wherein the object module comprises a prompt box control.

35. A method as in claim 17, wherein the object module comprises a pop-up menu control.

36. A method as in claim 17, wherein the object module comprises a check-box control.

37. A method as in claim 17, wherein the object module comprises a calendar selector control.

38. A method as in claim 17, wherein the object module comprises a number selector control.

39. A system for activating an object module in an application operating on a hypertext document containing a control element, the control element having attributes, comprising:

a station, the station hosts an application that operates on a hypertext document that contains a control element having an attribute;

a server, the server receives from the station the hypertext document that contains the control element having the attribute and sends to the station a GUI object for display in the hypertext document, the server having

an object module, the object module makes a GUI object of the attribute; and,

a software engine, the software engine parses the control element having the attribute in the hypertext document, sends the attribute to the object module, and invokes the object module to make the GUI object of the attribute.

40. A system for editing a hypertext document containing a control element, the control element having an attribute, comprising:

a station, the station hosts an application that operates on the hypertext that contains the control element having the attribute;

a server, the server receives from the station the hypertext document that contains a control element having at least one of an editable attribute and an edited attribute and sends to the station the editable attribute for display in an edit box, the server having

a set of object modules, at least one object module, makes a GUI object of the edited attribute; and,

a software engine, the software engine parses the

control element into a set of attributes to be edited, locates the object module corresponding to an edited attribute using a mapping from a location of the control element to a pointer to the object module corresponding to the edited attribute, and sends the edited attribute to the corresponding object module.